Computers Math. Applic. Vol. 26, No. 3, pp. 27–35, 1993
Printed in Great Britain.

0898-1221/93 $6.00 + 0.00

A New Look at Partial Fraction Expansion of Transfer Function Matrices from a Computational Viewpoint

J. LEYVA-RAMOS
Facultad de Ingeniería, Universidad Autónoma de San Luis Potosí
San Luis Potosí, S.L.P. 78290, Mexico

(Received April 1992; revised and accepted November 1992)

Abstract—This paper is concerned with the computation of residues of the partial fraction expansion of transfer function matrices. The main novelty of the procedure is: (1) all matrix residues are found at the same time; (2) no derivatives are required for repeated eigenvalues; and (3) there is no need for knowledge of the Jordan Canonical form. The procedure gives a general closed form for the partial fraction expansion and is particularly suitable for digital computer implementation with MATLAB. Numerical examples are included to illustrate the performance of the procedure.

1. INTRODUCTION

Consider the linear time-invariant, multivariable system described by

$$\frac{dx(t)}{dt} = A\,x(t) + B\,u(t), \tag{1a}$$

$$y(t) = C\,x(t), \tag{1b}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the control signal, and $y(t) \in \mathbb{R}^p$ is the vector of output measurements. The transfer function matrix of the dynamic system is

$$H(s) = C(s\,I_n - A)^{-1}\,B, \tag{2}$$

where $(sI_n - A)^{-1}$, called the resolvent matrix, is an $n \times n$ matrix, each element of which is a scalar transfer function, i.e., a ratio of two scalar polynomials.

It is often necessary to find the matrix residues of the partial fraction expansion (PFE) of transfer function matrices $(s\,I_n - A)^{-1}$ and $C(s\,I_n - A)^{-1}B$. There are several well-known methods available, namely:

(a) compute the $Adj(s\,I_n - A)^{-1}$ by minors or the Leverrier-Faddeeva algorithm and then perform a PFE on the result;
(b) by the use of the modal matrix, taking into consideration the particular nature of the Jordan form of A; and
(c) by the use of the function of a matrix (see [1–4]).

There are also other less known methods available:

(a) the Lagrange-Sylvester interpolation formula;
(b) the Cayley-Hamilton theorem; and
(c) Krylov's matrix (see [5–8]).

This work was supported in part by the Consejo Nacional de Ciencia y Tecnología, México. The author is thankful to the reviewer for his valuable comments.

The procedure given here is unique in the sense that it only uses a set of arithmetic operations, and it is particularly suitable for systems with repeated eigenvalues. The aforementioned procedure can easily be implemented in a digital computer with a very simple code and a MATLAB based implementation of this formula is included.

2. PARTIAL FRACTION EXPANSION OF THE RESOLVENT MATRIX

The resolvent matrix can be written into the two equivalent forms:

$$(s\,I_n - A)^{-1} = \frac{B_1\,s^{n-1} + B_2\,s^{n-2} + \cdots + B_{n-1}\,s + B_n}{s^n + d_1\,s^{n-1} + \cdots + d_{n-1}\,s + d_n} = \frac{B(s)}{d(s)}, \tag{3}$$

$$= \frac{B_1\,s^{n-1} + B_2\,s^{n-2} + \cdots + B_{n-1}\,s + B_n}{(s-\lambda_1)^{m_1}\,(s-\lambda_2)^{m_2}\,\ldots(s-\lambda_\sigma)^{m_\sigma}}, \tag{4}$$

where $B_i \in \mathbb{R}^{n\times n}$, λ_i is an eigenvalue of A and m_i its multiplicity, such that $n = \sum_{i=1}^{\sigma} m_i$, and σ is the number of distinct eigenvalues.

A third form is obtained by expanding into PFE

$$(s\,I_n - A)^{-1} = \sum_{i=1}^{\sigma}\sum_{j=1}^{m_i} \frac{F_{ij}}{(s-\lambda_i)^j}, \tag{5}$$

where the coefficients $F_{ij} \in C^{n\times n}$ are to be determined.

The representation given in (3) can also be expressed in polynomial form as $d(s)\,I_n = B(s)\,(s\,I_n - A)$, where matrix polynomials of order n are obtained on both sides. By comparing the coefficients of the matrix polynomials, the following expression is obtained:

$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} I_n & O_n & \ldots & \cdot & O_n \\ d_1\,I_n & \cdot & \ldots & \cdot & \cdot \\ \cdot & \cdot & \ldots & \cdot & \cdot \\ \cdot & \cdot & \ldots & \cdot & O_n \\ d_{n-1}\,I_n & \cdot & \ldots & d_1\,I_n & I_n \end{bmatrix} \begin{bmatrix} I_n \\ A \\ \vdots \\ A^{n-1} \end{bmatrix}, \tag{6}$$

where B_i are the matrix coefficients of the numerator polynomial of (3). To this end, recall the following well-known definition of Kronecker products:

DEFINITION 1. *Let X be an $n \times m$ matrix and Y be a $p \times q$ matrix. Then the $np \times mq$ matrix*

$$X \otimes Y = \begin{bmatrix} X_{11}\,Q & \ldots & X_{1m}\,Q \\ \cdot & \ldots & \cdot \\ \cdot & \ldots & \cdot \\ X_{n1}\,Q & \ldots & X_{nm}\,Q \end{bmatrix} \tag{7}$$

is called the Kronecker product of X and Y. It is easily verified from the above definition that $(X \otimes Y)\,(T \otimes S) = (X\,T \otimes Y\,S)$ when compatibility exists, see [9,10].

Thus, (6) is equivalent to the following Kronecker product representation:

$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = (D \otimes I_n) \begin{bmatrix} I_n \\ A \\ \vdots \\ A^{n-1} \end{bmatrix}, \tag{8}$$

where D is a lower Toeplitz matrix that contains the coefficients of $d(s)$.

The following formula for the matrix residues was derived in [11], which links (3) and (4):

$$\begin{bmatrix} F_{11} \\ F_{21} \\ \vdots \\ F_{n1} \end{bmatrix} = (V^{-1} D^{-1} \otimes I_n) \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix}, \tag{9}$$

where on the left-hand side are the desired matrix residues, and the matrix V is the Vandermonde matrix formed from the eigenvalues of A. For nonrepeated eigenvalues, this matrix is given by:

$$V = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \lambda_1 & \lambda_2 & \dots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \dots & \lambda_n^2 \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \dots & \lambda_n^{n-1} \end{bmatrix}. \tag{10}$$

The coefficients F_{ij} of (5) are obtained by combining (8) and (9); thus, it is seen that

$$\begin{bmatrix} F_{11} \\ F_{21} \\ \vdots \\ F_{n1} \end{bmatrix} = (V^{-1} \otimes I_n) \begin{bmatrix} I_n \\ A \\ \vdots \\ A^{n-1} \end{bmatrix}, \tag{11}$$

which is the desired result for obtaining the matrix residues of PFE of the resolvent matrix $(s\,I_n - A)^{-1}$.

For the multiple root case, the Vandermonde matrix in (10) is changed to the generalized Vandermonde matrix, defined as follows for an eigenvalue λ_i with multiplicity m_i:

$$V = \begin{bmatrix} 1 & \dots & 1 & 0 & 0 & \dots & 0 & \dots & 1 \\ \lambda_1 & \dots & \lambda_i & 1 & 0 & \dots & 0 & \dots & \lambda_\sigma \\ \lambda_1^2 & \dots & \lambda_i^2 & 2\lambda_i & 1 & \dots & 0 & \dots & \lambda_\sigma^2 \\ \cdot & \dots & \cdot & \cdot & \cdot & \dots & \cdot & \dots & \cdot \\ \cdot & \dots & \cdot & \cdot & \cdot & \dots & \cdot & \dots & \cdot \\ \lambda_1^{n-1} & \dots & \lambda_i^{n-1} & \binom{n-1}{1}\lambda_i^{n-2} & \binom{n-1}{2}\lambda_i^{n-3} & \dots & \binom{n-1}{m_i-1}\lambda_i^{n-m_i} & \dots & \lambda_\sigma^{n-1} \end{bmatrix}. \tag{12}$$

The result given in (11) shows that the residues are a linear combination of the powers of A, if $V^{-1} = W$; then, the combination for the i^{th} residue is:

$$F_{i1} = \omega_{i1}\, I_n + \omega_{i2}\, A_n + \cdots + \omega_{in}\, A^{n-1}. \tag{13}$$

3. PARTIAL FRACTION EXPANSION OF TRANSFER FUNCTION MATRICES

The matrix residues for the PFE of the transfer function matrix (2) can now be obtained by pre and post multiplication of (5) by C and B, i.e., the residues F_{ij}

$$C(s\,I_n - A)^{-1}\,B = \sum_{i=1}^{\sigma} \sum_{j=1}^{m_i} \frac{G_{ij}}{(s-\lambda_i)^j}, \tag{14}$$

then the residues in (14) become

$$G_{i1} = \omega_{i1}\, C\,B + \omega_{i2}\, C\,A\,B + \cdots + \omega_{in}\, C\,A^{n-1}\,B. \tag{15}$$

The above relation shows that the matrix residues are a linear combination of the Markov parameters [3], i.e., $h_i = C\,A^{i-1}\,B$ up to order $i = 1, \ldots, n$.

The expression given in (15) can be expressed in closed form as

$$\begin{bmatrix} G_{11} \\ G_{21} \\ \vdots \\ G_{n1} \end{bmatrix} = (V^{-1} \otimes I_p) \begin{bmatrix} C\,B \\ C\,A\,B \\ \vdots \\ C\,A^{n-1}\,B \end{bmatrix}, \tag{16}$$

or equivalently by:

$$\begin{bmatrix} G_{11} \\ G_{21} \\ \vdots \\ G_{n1} \end{bmatrix} = (V^{-1} \otimes C) \begin{bmatrix} B \\ A\,B \\ \vdots \\ A^{n-1}\,B \end{bmatrix}. \tag{17}$$

A rigorous proof which is completely different from the approach shown above can be obtained from the block observability realization as given in [12], or by the use of the well-known Leverrier-Faddeeva algorithm and the formula for PFE of matrix transfer functions [13].

Summary of the Procedure

The steps for computing the residues of the PFE of $H(s)$ can be stated as follows:

Step 1: Determine the set of eigenvalues of A with multiplicities m_i for $i = 1, 2, \ldots, \sigma$.

Step 2: Construct the Vandermonde matrix or the generalized Vandermonde matrix V using the structure given in (10) or (12).

Step 3: Use formula (16) or (17) to calculate the matrix residues.

4. DISCUSSION AND REMARKS ABOUT THE PROCEDURE

In this section, various properties and implementation issues associated with the procedure will be discussed.

(1) In this procedure, there is no need for knowledge of the minimal polynomial or the particular structure of the Jordan canonical form, and that is one of the most desirable properties of the procedure. The matrix A may have repeated roots, each one with the corresponding Jordan chain of eigenvectors. To calculate the specific structure may be very laborious with the associated numerical problems. In this procedure, if the order of the minimal polynomial is less than the corresponding multiplicity of the roots, then the residues of higher order will be zero.

(2) Unlike other methods, this procedure is direct, applying the corresponding formula; at the end, all the matrix residues are obtained simultaneously by matrix manipulation. It should be pointed out, that in order to obtain a high precision on the residues, a good program for computing the eigenvalues of the system matrix is necessary. The methods for solving the eigenvalue problem are iterative in nature and are, therefore, computationally expensive. Moreover, if a system is ill-conditioned, then the matrix residues computed using this approach may be inaccurate. For example, if the eigenvalues of A differ from each other by several orders of magnitude, then, while computing the inverse of the Vandermonde matrix, there could be significant loss of accuracy. A similar problem can occur while computing powers of A. In general, the accuracy of the matrix residues for this procedure will depend on the accuracy of the eigenvalues having about the same number of precision digits.

The procedure works directly from the eigenvalues of the matrix A; however, the user must be cautioned about the implementation of the algorithm. In Step 2, there is the

construction of the Vandermonde matrix. If the difference between two eigenvalues is very small, one could run into possible loss of accuracy or floating point overflow when the inverse is obtained, because two columns will be very similar; however, there is always the possibility of using doble precision.

(3) From the algorithm, it is clear that $n(n-2)$ operations are required to calculate the Vandermonde matrix for simple eigenvalues and $n(n-2)$ operations when the eigenvalue is repeated n times. Here, the standard definition for an operation as a multiplication and an addition will be used. For a matrix $n \times n$ there are different combinations of the multiplicity of the roots. Let us take the above figure for simplicity. The inverse of the Vandermonde will require n^3 operations approximately, then for the PFE of the resolvent matrix, it will take $n^3(n-2)$ to obtain the matrix with the powers of A and an additional n^3 to perform the multiplication in the formula, so the total account will be $n^4 + n(n-2)$, which is approximately n^4. For the residues of the transfer function matrix, it will take approximately $n^2\,p(m^2+n)$, where the bulk of the computational effort is involved in calculating the column block matrix containing the powers of A that multiplies the matrix $(V^{-1} \otimes C)$.

(4) The use of complex matrices can be avoided by separating the real and imaginary parts of the Vandermonde matrix V. Then, an isomorphism can be used:

$$V = V_R + j\,V_I \leftrightarrow \begin{bmatrix} V_R & V_I \\ -V_I & V_R \end{bmatrix}. \tag{18}$$

This new matrix contains the same spectrum as V plus the conjugate spectrum. The required inverse can be obtained from this matrix where the isomorphism is preserved. The upper left and right blocks will contain the real and imaginary parts of the inverse, respectively. Then, formula (17) can be modified using kronecker products as

$$(V^{-1} \otimes C) = (W_R \otimes C + j\,W_I \otimes C).$$

Therefore, the real and imaginary parts of the matrix residues can be obtained independently.

A listing of a MATLAB [14] version of the algorithm is included here. The input to the program is the triplet (A, B, C) with the corresponding multiplicity. For the application of the formula, the built-in functions *inv* and *kron* are used to simplify the procedure. The output of the program, namely, the matrix residues of the expansion, will appear in increasing order in array *res*. The numerical results given in the next section were computed using this program.

5. EXAMPLES

Here, two numerical examples are given to illustrate the performance of the procedure.

EXAMPLE 1. Consider the system matrix A given by

$$A = \begin{bmatrix} -3 & -4 & -4 & -2 & -1 \\ 7 & 8 & 7 & 4 & 2 \\ -3 & -2 & -2 & -3 & -2 \\ -3 & -4 & -2 & 2 & 1 \\ 11 & 10 & 10 & 5 & 4 \end{bmatrix}. \tag{19}$$

We would like to compute the matrix residues of PFE of the resolvent matrix $(s\,I_n - A)^{-1}$.

Here, $det(s\,I_n - A) = (s-1)\,(s-2)^4$; therefore, there are two distinct eigenvalues, $\lambda_1 = 1$ with $m_1 = 1$ and $\lambda_2 = 2$ with $m_2 = 4$. In this example, $nullity(\lambda_2\,I_5 - A) = 2$, then there exist two

independent Jordan chains for the eigenvalue $\lambda_1 = 2$. Thus, the Jordan form for the matrix A is:

$$J = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}.$$

Because there are repeated eigenvalues, the structure for the generalized Vandermonde matrix as given in (12) has to be used:

$$V = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ \lambda_1 & \lambda_2 & 1 & 0 & 0 \\ \lambda_1^2 & \lambda_2^2 & 2\lambda_2 & 1 & 0 \\ \lambda_1^3 & \lambda_2^3 & 3\lambda_2^2 & 3\lambda_2 & 1 \\ \lambda_1^4 & \lambda_2^4 & 4\lambda_2^3 & 6\lambda_2^2 & 4\lambda_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 1 & 4 & 4 & 1 & 0 \\ 1 & 8 & 12 & 6 & 1 \\ 1 & 16 & 32 & 24 & 8 \end{bmatrix}.$$

Now, using the formula (11) the matrix residues can be obtained from:

$$\begin{bmatrix} F_{11} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{24} \end{bmatrix} = (V^{-1} \otimes I_n) \begin{bmatrix} I_5 \\ A \\ A^2 \\ A^3 \\ A^4 \end{bmatrix}.$$

By substituting values of V and A, the multiplication gives the matrix residues in the proper order. The complete PFE of the transfer function matrix is, therefore, given by

$$(s\,I_5 - A)^{-1} = \frac{\begin{bmatrix} 4 & 2 & 2 & 1 & 1 \\ -4 & -2 & -2 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 4 & 2 & 2 & 1 & 1 \\ -8 & -4 & -4 & -2 & -2 \end{bmatrix}}{(s-1)} + \frac{\begin{bmatrix} -3 & -2 & -2 & -1 & -1 \\ 4 & 3 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ -4 & -2 & -2 & 0 & -1 \\ 8 & 4 & 4 & 2 & 3 \end{bmatrix}}{(s-2)} + \frac{\begin{bmatrix} -1 & -2 & -2 & -1 & 0 \\ 3 & 4 & 5 & 3 & 1 \\ -3 & -2 & -4 & -3 & -2 \\ 1 & -2 & 0 & 1 & 2 \\ 3 & 6 & 6 & 3 & 0 \end{bmatrix}}{(s-2)^2}, \tag{20}$$

which is the correct expansion of $(s\,I_5 - A)^{-1}$. Also, it is important to notice that $F_{23} = O_{5\times5}$ and $F_{24} = O_{5\times5}$; that is because the minimal polynomial of A is $\varphi = (s-1)(s-2)^2$ as can be seen from the corresponding Jordan form.

When the correct eigenvalues of A were used in the program, the matrix residues F_{11}, F_{21}, and F_{22} were accurate up to 14 significant digits; and for the remaining residues F_{23} and F_{24} the elements are multiplied by 10^{-13}. When MATLAB calculates the eigenvalues of A using the built-in function *eig*(A), it gives the following values: $\lambda_1 = -1.00000000000001$, $\lambda_2 = -2.00000004885672$, $\lambda_3 = -1.99999995114327$, $\lambda_4 = \lambda_5 = -1.99999999999999 \pm i0.00000012307120$ which shows that the accuracy for the eigenvalue λ_2 is up to 8 significant digits. When the values of λ_1 with multiplicity $m_1 = 1$ and λ_2 with multiplicity $m_2 = 4$ are used in the generalized Vandermonde matrix, the accuracy for each matrix residue is as follows: F_{11} with 13 significant digits, F_{21} with 13 significant digits, and F_{22} with 8 significant digits. For F_{23}, the elements are multiplied by 10^{-6}

and for F_{24} by 10^{-12}. MATLAB uses EISPACK routines which are standard for computing the eigenvalues.

EXAMPLE 2. Consider the 3rd order system described by:

$$\frac{dx(t)}{dt} = \begin{bmatrix} 1 & -1 & 0 \\ 3 & -4 & 1 \\ 5 & -6 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 2 \\ 0 & 3 \\ 1 & 0 \end{bmatrix} u(t), \tag{21a}$$

$$y(t) = \begin{bmatrix} 1 & 5 & 0 \\ 4 & 1 & 2 \end{bmatrix} x(t). \tag{21b}$$

Here, $det(sI - A) = s(s + 1 \pm i1)$; therefore, the three eigenvalues are distinct with two complex conjugates. Then the matrix residues will be complex for the complex eigenvalues. The Vandermonde matrix has its simplest form (10), and upon applying the formula (17), the matrix coefficients of the PFE can be obtained from

$$\begin{pmatrix} G_{11} \\ G_{21} \\ G_{31} \end{pmatrix} = \left(\begin{bmatrix} 1 & 1 & 1 \\ 0 & -1+i & -1-i \\ 0 & -2i & 2i \end{bmatrix}^{-1} \otimes \begin{bmatrix} 1 & 5 & 0 \\ 4 & 1 & 2 \end{bmatrix} \right) \begin{pmatrix} B \\ AB \\ A^2B \end{pmatrix}.$$

Carrying out the multiplication gives the expansion as

$$H(s) = \frac{\begin{bmatrix} 3.0000 & 21.0000 \\ 3.5000 & 24.5000 \end{bmatrix}}{s} + \frac{\begin{bmatrix} -1.0000 & -2.0000 \\ 1.2500 & -6.7500 \end{bmatrix} + i \begin{bmatrix} -9.5000 & 17.5000 \\ -11.2500 & 19.7500 \end{bmatrix}}{s+1-i} + \frac{\begin{bmatrix} -1.0000 & -2.0000 \\ 1.2500 & -6.7500 \end{bmatrix} - i \begin{bmatrix} -9.5000 & 17.5000 \\ -11.2500 & 19.7500 \end{bmatrix}}{s+1+i}, \tag{22}$$

which is the correct expansion of $H(s) = C(sI - A)^{-1}B$. Because complex numbers are allowed in all the operations and functions in MATLAB, the program can compute complex matrix residues too, as shown in this example.

6. CONCLUSIONS

A computational procedure is given for obtaining the matrix residues of PFE of transfer function matrices. The procedure is direct and computes all the residues at the same time and no knowledge is necessary of the particular nature of the Jordan form of the system matrix A. This is a very desirable advantage of the procedure. It only involves operations on numerical matrices where the Vandermonde matrix V is formed from the eigenvalues of A. This procedure is particularly suitable for those systems of high order, with multiple eigenvalues. When MATLAB is used, because of the built-in functions, this procedure becomes very powerful.

APPENDIX

The algorithm for the procedure was implemented using MATLAB. The implementation shown below uses the built-in functions: *eig*, for calculating the eigenvalues of the system matrix A; *inv*, for the inversion of the Vandermonde matrix; and *kron*, for the kronecker product between the inverse of the Vandermonde matrix and C. The user can generate the code for any other programming language, using the guidelines in the listing below.

```
echo off
%
% This program computes the matrix residues of the partial
% fraction expansion of H(s) = C(s I_n - A)^-1 B.
```

```
%
%          A = matrix of dimensions n × n
%          B = matrix of dimensions n × p
%          C = matrix of dimensions m × n
%          V = Vandermonde matrix
%       root = a vector containing the nonrepeated eigenvalues of A
%     porder = a vector representing the order of each corresponding eigenvalue.
%        res = the matrix residues of the expansion.
%
n = input('type the order of the system matrix A =');
p = input('type the number of columns of matrix B =');
m = input('type the number of rows of matrix C =');
A(1 : n, 1 : n) =input('type the matrix A =')
B(1 : n, 1 : p) =input('type the matrix B =')
C(1 : m, 1 : n) =input('type the matrix C =')
e = eig(A);
eps = 1.0e − 6; sigma = 0; i = 1;
while i <= n
      sigma = sigma + 1;
      root(sigma) = e(i);
      porder(sigma) = 1;
      for j = i + 1 : n,
      if abs(real(e(i)) − real(e(j))) < eps & abs(imag(e(i)) − imag(e(j))) < eps
            dummy = e(j); e(j) = e(i); e(i) = dummy;
            porder(sigma) = porder(sigma) +1;
            i = i + 1;
            end
      end
      i = i + 1;
end
%
% Construct the Vandermonde matrix from the eigenvalues of A
%
r = 0;
for k = 1: sigma,
      r = r + 1; van(1,r) = 1.0;
      for i = 2 : n,
            van(i, r) = van(i − 1, r)*root(k);
      end
      j = 2;
      while porder(k) >= j
            r = r + 1;
            for i = 1 : n,
                  if i < j,
                        van(i, r) = 0;
                  else if i == j,
                        van(i, r) = 1;
                  else
                        van(i, r) = van(i − 1, r − 1)*(i − 1)/(j − 1);
                  end
            end
            j = j + 1;
      end
end
%
```

```
% Calculate the residues using the new formula
%
b(1 : n, 1 : p) = B;
for i = 1 : n - 1
    b(i*n + 1 : (i + 1)*n, :) = A*b((i - 1)*n + 1 : i*n, :);
end
res = kron(inv(van),C)*b;
disp('the residues of the expansion are =')
r = 0;
for k = 1:sigma
    fprintf('\nthe residues for the eigenvalue %g %gi \n\n',real(root(k)),imag(root(k)))
    for i = 1:porder(k)
        r = r + 1;
        fprintf(' factor = %2.0f\n\n',i)
        disp(res((r - 1)*m + 1 : r*m,:))
    end
end
```

REFERENCES

1. C.T. Chen, *Linear System Theory and Design*, Holt, Rinehart and Winston, New York, (1984).
2. R.A. DeCarlo, *Linear Systems: A State Variable Approach with Numerical Implementation*, Prentice-Hall, Englewood Cliffs, NJ, (1989).
3. T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ, (1980).
4. L.A. Zadeh and C.A. Desoer, *Linear System Theory*, McGraw-Hill, New York, (1963).
5. C.F. Chen and G. Freeman, A new formula for partial fraction expansion of a transfer matrix, In *Proc. Amer. Contr. Conf.*, (1986).
6. C.F. Chen, A new approach to matrix Heaviside expansion, *Int. J. Control* **3**, 431–448 (1970).
7. F.R. Gantmacher, *The Theory of Matrices*, Chelsea, New York, (1959).
8. H. Rosenbrock, Partial fraction expansion of a transfer matrix, *Electron. Lett.* **1** (1966).
9. R. Bellman, *Introduction to Matrix Analysis*, McGraw-Hill, New York, (1970).
10. J.W. Brewer, Kronecker products and matrix calculus in system theory, *IEEE Trans. Circuits Syst.* **9**, 772–781 (1981).
11. J. Leyva-Ramos and E.D. Denman, Matrix partial-fraction expansion method, *Electron. Lett.* **3**, 99–100 (1983).
12. J. Leyva-Ramos, Partial fraction expansion in system analysis, *Int. J. Control* **3**, 619–639 (1991).
13. J. Leyva-Ramos, A method for partial fraction expansion of transfer matrices, *Trans. Automat. Contr.* **12**, 1472–1475 (1991).
14. The MathWorks, Inc., *MATLAB for Macintosh Computers*, Natick, MA, (1991).